

Orientation estimation for instrumented helmet using neural networks

Measurement and Control
2023, Vol. 56(7-8) 1156–1167
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/00202940221149062
journals.sagepub.com/home/mac

Muhammad Hamad Zaheer¹, Se Young Yoon¹ ,
and Brian K Higginson²

Abstract

This work presents an integrated solution for head orientation estimation, which is a critical component for applications of virtual and augmented reality systems. The proposed solution builds upon the measurements from the inertial sensors and magnetometer added to an instrumented helmet, and an orientation estimation algorithm is developed to mitigate the effect of bias introduced by noise in the gyroscope signal. Convolutional Neural Network (CNN) techniques are introduced to develop a dynamic orientation estimation algorithm with a structure motivated by complementary filters and trained on data collected to represent a wide range of head motion profiles. The proposed orientation estimation method is evaluated experimentally and compared to both learning and non-learning-based orientation estimation algorithms found in the literature for comparable applications. Test results support the advantage of the proposed CNN-based solution, particularly for motion profiles with high acceleration disturbance that are characteristic of head motion.

Keywords

Orientation estimation, attitude estimation, inertial measurement unit, machine learning, convolutional neural networks, sensor fusion

Date received: 29 June 2022; accepted: 24 November 2022

Introduction

Accurate measurement of head orientation plays a critical role in providing seamless experiences in virtual reality (VR) and augmented reality (AR) applications.¹ Accurate tracking of head motion and orientation is needed in VR and AR systems to correctly place digital objects relative to the real physical world. For example, AR technology incorporated to the equipment for first responders, such as goggles and helmets, can be used to deliver safety-critical information in real-time, with digital markers that are anchored to physical objects in a person's field of vision. Head orientation data is also used in health and safety applications, which include the monitoring of fatigue levels and other mental/physical changes in workers that may lead to accidents and injuries.² The study presented in the current paper is motivated by the goal of achieving accurate and precise head orientation estimation using measurements from an instrumented helmet.

Infrared cameras and optical flow sensors are among the most accurate solutions in tracking the attitude of rigid bodies. On the downside, the elaborate hardware setups and enclosed environment that these systems need restrict their portability and usage in

uncertain environments. Recently, advances in micro-electromechanical systems (MEMS) have led to the development of miniaturized Inertial Measurement Units (IMUs), and their accessibility and packaging have made them a popular solution for orientation estimation systems in VR and AR applications with high mobility requirement.^{3,4} A standard 9-axis IMU provides measurements of the magnetic field, angular velocity, and acceleration from the integrated magnetometer, gyroscope and accelerometer sensors, respectively.

Measurement noise represents a recurring challenge in all sensor implementations, and IMUs are also susceptible to these external disturbances. For example, measurement noise from a gyroscope can accumulate to cause a drift in the orientation estimate away from

¹Department of Electrical and Computer Eng., University of New Hampshire, Durham, NH, USA

²Warfighter Systems Integration Lab, Galvion Ltd, Portsmouth, NH, USA

Corresponding author:

Se Young Yoon, Department of Electrical and Computer Eng., University of New Hampshire, Kingsbury Hall W201, Durham, NH 03824, USA.
Email: seyoung.yoon@unh.edu



the true values. Solutions to mitigate this long-standing challenge in orientation estimation include methods that leverage multiple vector measurements in the inertial and the body frame to estimate orientation.⁵ In the case of satellites, these vector measurements can be the direction vectors of the sun and other stars,⁶ while in robotics and consumer electronics the gravity vector and magnetic north vector may be used.^{7,8} By combining and fusing two or more estimates of orientation, the objective of these solutions is to mitigate the sensitivity to noise of individual sensor units and to provide a more accurate estimation of orientation. A small sample of the different orientation estimation algorithms found in the literature is discussed in the next section.

Related works

Orientation estimation algorithms can be broadly divided into two groups based on whether learning-based approaches were used during their design or the design was purely based on a model-based approach.

Non-learning-based methods

Non-learning-based orientation estimation methods can be divided into two categories: Bayes filter and complementary filter. Bayes filter leverages the dynamic model of the moving rigid body and the probability characteristics of the measurement noise to iteratively calculate the most likely orientation. The simplest form of the Bayes filter under a linear observable dynamic model and additive Gaussian measurement noise is the Kalman Filter,⁹ while the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF)¹⁰ offer approximated solutions to applications with nonlinear models. Error State Kalman Filter (ESKF) offers an improved alternative to EKF in certain applications by estimating the expected error in orientation calculation. In particular, ESKF has been shown to provide better results than EKF in estimating aircraft attitude¹¹ and mobile robot actuator orientation.¹²

Complementary filters determine the orientation of a rigid body by combining through a weighted average multiple orientation estimates obtained from different observations.¹³ In an IMU, these observations may originate from the gyroscope's angular velocity measurement, or vector measurements of gravity and magnetic field. TRIAD,⁵ QUEST,⁵ Fast Optimal Matrix Algorithm (FOAM),¹⁴ Optimal Linear Attitude Estimator (OLAE),¹⁵ Fast Linear Attitude Estimator (FLAE)¹⁶ are some examples of orientation estimation solutions from vector measurements. The optimal weights combining the observations in a complementary filter are selected heuristically based on the observed noise levels and characteristics on the input measurements.^{13,17} Lower weights are assigned to noisy measurements within the time or frequency ranges where the noise observations are predominant, while

the remaining input weights are adjusted to maintain the weighted average of the filter output.

Learning-based methods

Learning-based approaches have been used to improve the performance of traditional orientation estimation filters. Support Vector Regression (SVR)¹⁸ has been studied in combination with Kalman filters in Yan et al.¹⁹ to detect the properties of motion profiles and tune the filter parameters for best performance, while^{20–22} use Recurrent Neural Network (RNN) for the filter tuning process. Artificial Neural Network (ANN) is considered in Chiang et al.²³ for smoothing the output of Kalman filters and improving the accuracy of the orientation estimate.

In more recent works, neural networks have been used to identify black box models for end-to-end orientation estimation. It is shown in Weber et al.²⁴ that these neural network models based on Recurrent Neural Networks (RNNs) and Temporal Convolutional Networks (TCNs) can outperform non-learning-based methods in some applications. A simple Convolutional Neural Network (CNN) model²⁵ with reduced number of layers and fast training time is evaluated for applications with quadcopters in Brossard et al.,²⁶ where the model is used to denoise gyroscope measurements. A Long Short Term Memory (LSTM) based model is proposed in Esfahani et al.²⁷ for estimating changes in quadcopter orientation, where the model uses gyroscope measurements to estimate the change in orientation at a constant sampling time. For applications where the sampling rates are varying over time, Esfahani et al.²⁸ proposes a modified LSTM-based model that incorporates the sampling rate information as an input to the model. Bidirectional-LSTM (Bi-LSTM), which extracts information from the training data in both the forward and backward direction, is considered for vehicle odometry applications in Zhao et al.,²⁹ where it is shown to improve the estimation results when compared to the standard LSTM. Table 1 summarizes some of these recent works on learning-based methods for orientation estimation and compares them with our proposed method.

While the integration of machine learning models to orientation estimation methods has been shown to improve accuracy in many application domains, the capabilities of these solutions in applications related to head orientation estimation remain largely unknown. Previous works on head orientation estimation using neural networks, such as,^{30,31} have mainly focused on solutions that rely on video and images data captured by carefully configured cameras, which limit their applicability in situations requiring high mobility and outdoor usage. Head orientation in VR and AR applications also follow motion profiles that are unique to the activities during which the head movement is measured, and the availability of sensor data varies from comparable applications in which IMUs are used for

Table 1. Machine learning-based methods for orientation estimation.

Paper	Method	Application	Type
Proposed	CNN	Head O ^a	E ^c
Chiang et al. ²³	ANN	Vehicle N ^b	A ^d
Yan et al. ¹⁹	SVR	Pedestrian N	A
Wagstaff and Kelly ²¹	LSTM	Pedestrian N	E
Brossard et al. ²²	LSTM	Mobile robot N	A
Esfahani et al. ²⁸	LSTM	Quadcopter O	E
Esfahani et al. ²⁷	LSTM	Quadcopter N	E
Weber et al. ²⁴	RNN	Quadcopter O	E
Brossard et al. ²⁶	CNN	Quadcopter O	E
Herath et al. ⁴	LSTM	Pedestrian N	E
Zhao et al. ²⁹	Bi-LSTM	Vehicle N	E

^aOrientation is estimated.

^bNavigation: Orientation and position are estimated.

^cEnd-to-end: Standalone machine learning model is used.

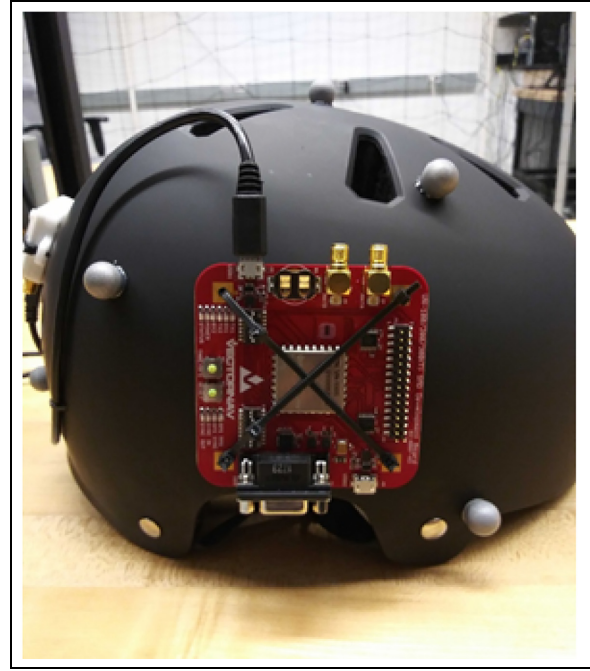
^dAugmented: Machine learning is used to enhance non-learning-based methods.

attitude estimation. For example, much of the previous works with neural network models have been trained and evaluated using IMU datasets such as EuRoC³² and TUM VI,³³ which are catered to quadcopter flight maneuvers and include motion profiles with lower peak angular velocities than the head motions we consider in our current work. Additionally, neural network models intended for quadcopter applications do not use magnetometer measurements due to the high level of noise that is injected by the motors. This noise can lead to greater drift and error in estimation of yaw angle.

Contributions of this work

Driven by the above-described need for learning-based solutions that can provide an accurate estimation of head orientation, this paper presents a new CNN-based solution that dynamically estimates head orientation from the 9-axis IMU measurements of an instrumented helmet. For the design and evaluation of our proposed solution, a new IMU dataset specific to head motion profiles is collected as part of this study. The new dataset is leveraged in the design of the proposed estimation solution to achieve a more accurate estimate of orientation in applications with high angular velocities and accelerations. A comprehensive evaluation of the proposed solution is provided in this work, including performance comparisons to benchmarks set by non-learning and learning-based algorithms in the literature.^{26,27} The main contributions of this paper are then summarized as follows.

- A new method for estimating head orientation is proposed, which is inspired by complementary filters, and incorporates a CNN model that dynamically estimates orientation from the 9-axis measurements of IMUs.

**Figure 1.** Helmet with vectorNav IMU and markers for the motion capture system.

- A comprehensive evaluation of the proposed solution is presented here, and its performance in estimating head orientation is compared to benchmarks set by established learning and non-learning solutions found in the literature.
- The HELMET dataset is introduced, which is intended to capture how human head moves while performing various physical activities. Test results are presented to identify the unique characteristics observed in the head motion data, and to verify the effect that these characteristics have on the expected performance of the orientation estimation methods.

Problem description

The objective of this work is to design an integrated solution that can accurately estimate head orientation. The measurements needed for the estimation come from an IMU with a magnetometer rigidly attached to a helmet as shown in Figure 1. The aim is to determine the orientation of the helmet, which is measured by the rotation of the body frame B (Figure 2) relative to the inertial frame I (Figure 3), and in the presence of external disturbances and sensor noise.

The orientation of the helmet can be obtained from the gyroscope sensor of the IMU, which provides measurements of angular velocities about the body frame. By integrating the gyroscope's angular velocity measurements $\tilde{\omega}_t$ over time t , an estimate of orientation $q_{\omega,t}$ at time t can be obtained. Another estimate of orientation $q_{\nabla,t}$ can be obtained by from the combined vector $\tilde{u}_t = [\tilde{a}_t \quad \tilde{m}_t]$ of accelerometer $\tilde{a}_t$ and magnetometer

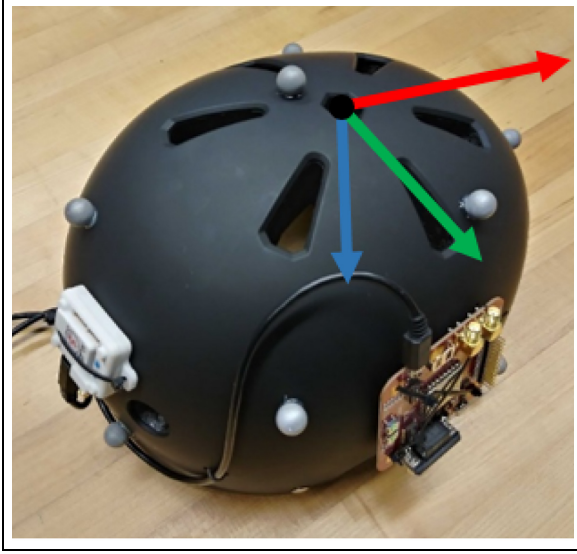


Figure 2. The helmet coordinate frame B. Red is the x-axis, Green is the y-axis and Blue is the z-axis.

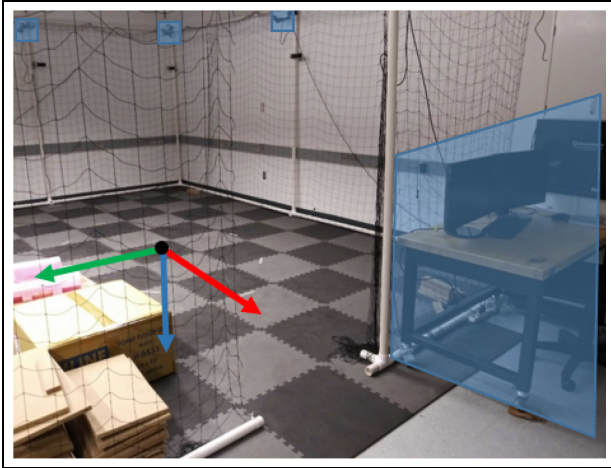


Figure 3. Fixed inertial coordinate frame shown relative to the test floor. A motion capture system consisting of the ground station and infra-red cameras is used to record the orientation of the helmet, which is used as the ground truth during training.

measurements $\tilde{m}_t$ and using vector-based estimation methods like QUEST.⁵

As the orientation estimate $q_{\omega,t}$ is susceptible to high bias error for noisy gyroscope measurements, while high-frequency noise and disturbances in the accelerometers and magnetometers measurements propagate to the orientation estimate $q_{\nabla,t}$, a complementary filter as shown in Figure 4 can be used to combine the two estimates of orientation to improve accuracy,

$$q_{c,t} = \alpha q_{\omega,t} + (1 - \alpha) q_{\nabla,t}. \quad (1)$$

The properties of the combined orientation estimate $q_{c,t}$ depend on the selection of α , which can be static or

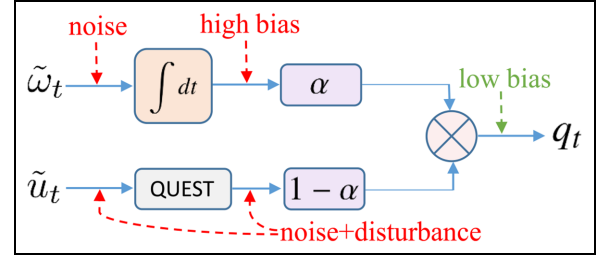


Figure 4. Schematic illustration of the complementary filter. The quaternion obtained from integrating the gyroscope measurements contains high bias, while the quaternion from QUEST contains disturbances and noise. The two are combined to decrease noise and bias.

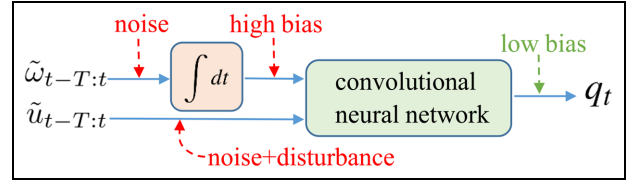


Figure 5. Schematic illustration of the proposed method. Convolutional Neural Network is used to remove bias from quaternion calculated by integrating gyroscope measurements.

time-varying. The function α is commonly selected heuristically based on the motion profile of the target body, as well as the levels of noise and disturbances introduced to the sensor measurements.

Filter tuning through machine learning

One way to automate the selection of the function α in complementary filters (1) is to use machine learning. The value of such α may be determined as a function of the estimated rotation quaternions within a local window of size T as,

$$\alpha(t) = g(q_{\omega,t}, \dots, q_{\omega,t-T}, q_{\nabla,t}, \dots, q_{\nabla,t-T}), \quad (2)$$

where $g(\cdot)$ can be identified using machine learning methods such as CNN. Further integration of machine learning can replace the QUEST algorithm to find the rotation quaternion $q_{\nabla,t}$, and instead feed the accelerometer and magnetometer measurements directly to the CNN model. Therefore, a proposed CNN architecture as shown in Figure 5 can combine the quaternions obtained from the gyroscope with accelerometer and magnetometer measurements to estimate the rotation quaternion corresponding to the current head orientation as,

$$q_t = f(q_{\omega,t}, \dots, q_{\omega,t-T}, \tilde{u}_t, \dots, \tilde{u}_{t-T}), \quad (3)$$

where $f(\cdot)$ is the function implemented by the CNN.

Neural network structure

CNN is used for the network architecture in Figure 5 as it results in a less complex network compared to RNN and LSTM, which facilitates training and implementation in practical applications. The sliding convolution filters of the CNN also prevent the overfitting data in the identification of the trained model (3) by taking advantage of the time structure of the input data and promoting connections between input data samples that are closer in time.

As the performance of neural networks is highly dependent on the hyper-parameters of the model, different combinations of hyper-parameters were evaluated in this study for the application of interest. Some of the hyper-parameters of the neural network like learning rate, learning rate drop factor, number of neurons in each layer, and depth of the network were selected using Bayesian optimization,³⁴ which is a probabilistic estimation method that models the hyper-parameter function using Gaussian processes and then finds the minimum of the estimated functions. Batch normalization was used between linear layers, which improves the trainability of deep neural networks without overfitting the resulting model to the training data.³⁵ Other hyper-parameters of the neural network such as the activation functions between layers, the loss function, and the optimizer were selected by trial-and-error as to be discussed in the next section. We also used data augmentation to avoid over-fitting the neural network on the training data.

Activation functions

Activation functions are used between linear layers to introduce nonlinearities to the neural network model. The accuracy and generalizability of the neural network model depend on the activation functions used. Sigmoid, Gaussian Error Linear Unit (GELU), rectified linear (ReLU) and tanh are three of the most commonly used activation functions.³⁶ Recently, activation functions with learnable parameters like Parametric Rectified Linear Unit (PReLU) have achieved better results in different domains.³⁷ As the output of the CNN in the current application is in quaternion with elements that are bounded by $[-1, 1]$, the tanh activation function is an intuitive choice for the last layer. Different combinations of activation functions were evaluated for the remaining layers to achieve the best estimation performance, and the corresponding test results are summarized in Section 5.

Loss function

The loss function is another hyper-parameter that determines the performance of neural networks. In the current application, the loss function quantifies the closeness of the rotation quaternion estimated by the CNN model to the quaternion corresponding to

the true orientation. In many artificial neural network applications, mean square loss (MSL) is an effective loss function, but for the current model MSL fails to take into account the special structure of the rotation quaternion and it was found to be inadequate. On the other hand, Quaternion Angle Error (QAE)³⁸ has been shown to provide a more reliable quantification of the closeness between two quaternions, and it is broadly used to evaluate the accuracy of attitude estimation algorithms. Considering rotation quaternions as 4-dimensional unit vectors, QAE is calculated as

$$\theta_e = \cos^{-1}(2(q^T \tilde{q})^2 - 1), \quad (4)$$

where q is the true rotation quaternion and $\tilde{q}$ is the estimated rotation quaternion. It is noted that θ_e is zero when the two rotation quaternions are equal, and its magnitude increases with the angle between the quaternions.

A simpler approximation of the QAE can be obtained by noting that the angle between two unit vectors q and $\tilde{q}$ is given by $\cos^{-1}(q^T \tilde{q})$, which quantifies the angular separation between the vectors. Therefore, a loss function L may be defined as,

$$L^{(1)} = \cos^{-1}(q^T \tilde{q}), \quad (5)$$

for a pair of estimated rotation quaternions $\tilde{q}$ and the corresponding true rotation quaternion q . The gradient of the above function grows to infinity as the vector product approaches one (e.g. for parallel and anti-parallel quaternions).²⁴ To avoid this, the first or second-order Taylor approximations of the inverse cosine function, given as

$$\cos^{-1}(x) \approx \pi/2 - x, \quad (6)$$

and

$$\cos^{-1}(x) \approx \pi/2 - x - x^3/6, \quad (7)$$

may be used to implement the loss function (5).

The proximity of two rotation quaternions can also be quantified by the quaternion product. The product of a quaternion and its conjugate is equal to the unit quaternion, and thus alternative loss functions quantifying the difference between a pair of estimated and true rotation quaternions can be defined as

$$L^{(2)} = |q \otimes \tilde{q} - [1 \ 0 \ 0 \ 0]^T|, \quad (8)$$

where $|\bullet|$ is the $\mathcal{L}_1$ vector norm and $\otimes$ is the quaternion product operator, or using the $\mathcal{L}_2$ vector norm $\|\bullet\|$ as

$$L^{(3)} = \|q^* \otimes \tilde{q} - [1 \ 0 \ 0 \ 0]^T\|. \quad (9)$$

Let $L_{(i)}^{(j)}$, be the loss of the i^{th} sample of the estimated quaternion calculated using the loss function $L^{(j)}$ as defined in (5), (8), or (9). The total loss corresponding to N quaternion estimate samples is defined in the current application as,

$$\mathbf{L}_1 = \sum_{i=1}^N |L_{(i)}^{(j)}|, \quad (10)$$

or alternatively by

$$\mathbf{L}_2 = \sqrt{\sum_{i=1}^N \left(L_{(i)}^{(j)}\right)^2}. \quad (11)$$

Combinations of the loss functions (5), (8), (9) and the total loss (10), (11), are implemented in training of the CNN model for the current application, and a comparison of the trained models' performances is presented in Section 5.

Optimizer

Optimizers are algorithms used in the training of neural networks to aid in the selection of model parameters that minimize the values of the loss functions. A combination of RAdam³⁹ and Lookahead⁴⁰ was used to train the proposed neural network model. Both RAdam and Lookahead methods have been proven to be effective at optimizing the learning process in machine learning methods, including in applications of orientation estimation.²⁴

Data augmentation

Data augmentation⁴¹ is a technique used in machine learning to significantly increase the amount of available training data. The augmented data also acts as a regularizer and helps to avoid overfitting the trained model. For orientation estimation, some methods of data augmentation are adding Gaussian noise, adding static bias, and rotating the IMU and magnetic measurements.^{26,28} The first method is used commonly in machine learning applications and involves adding Gaussian noise to the input data which helps avoid overfitting by forcing the neural network to learn a more general relation between the input and the output training data. Static bias in the accelerometer and magnetometer measurements can appear due to different operating conditions of the IMU like temperature and calibration errors. As this should not cause the orientation estimate to be affected, we can generate accelerometer and magnetometer data with different biases to improve the robustness of the neural network to sensor bias. Similarly, rotating the IMU measurements and the corresponding ground truth orientation provides additional data for training.

The CNN-based model proposed in Figure 5 requires an initial quaternion value for the integration step. This initial value of the integrator is calculated using the QUEST algorithm, which is sensitive to acceleration and magnetic disturbances. As part of the data augmentation process, we can add small variations to this initial quaternion estimate to account for these type of disturbances.

Experimental results

The training and evaluation of the head orientation estimation method proposed in Section 4 are presented here. The CNN model is trained using a dataset of head motion profiles collected as part of this study, and different combinations of training hyper-parameters are evaluated. The performance of the proposed CNN model is also compared to established learning and non-learning-based algorithms proposed in the literature for orientation estimation using 9-axis IMUs, such as the ESKF with magnetic angular rate update⁴² and complementary filter with gyroscope bias tracking and disturbance estimation (to be referred as Mahony et al.).¹³ We also compared our method to the CNN-based signal denoising method in Brossard et al.²⁶

Datasets

In order to train and evaluate the CNN model with motion profiles that are relevant to head motion, a dataset (HELMET) of IMU measurements was collected using the instrumented helmet in Figure 1. The inertial and magnetic data is collected using a VectorNav IMU module,⁴³ which contains a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. The VectorNav module is rigidly attached to the helmet, which also has eight reflective markers distributed on its outer surface as shown in Figure 1. The position of the markers allows an OptiTrack infrared motion capture system⁴⁴ shown in Figure 3 to track the motion and orientation of the helmet with high precision. The measured helmet orientation by the OptiTrack system defines the true rotation between the helmet frame in Figure 2 and the inertial frame in Figure 3. The inertial (acceleration, angular velocity) and magnetic data collected from the IMU serve as the input to the CNN model during training and testing, and the orientation determined by the motion capture system is used as the ground truth.

Twelve sets of motion profile data were collected for the dataset, each of which is 300 s in length and sampled at 250 Hz. The helmet motion data was collected while performing different dynamic activities to cover a wide range of possible motions. Sets numbered 1–4 were recorded for head movements in a sitting position, sets 5–8 were recorded while walking at a slow and medium pace and sets 9–12 were recorded while running with sudden stops. An example of head motion trajectories captured during the recording of the data sets is given

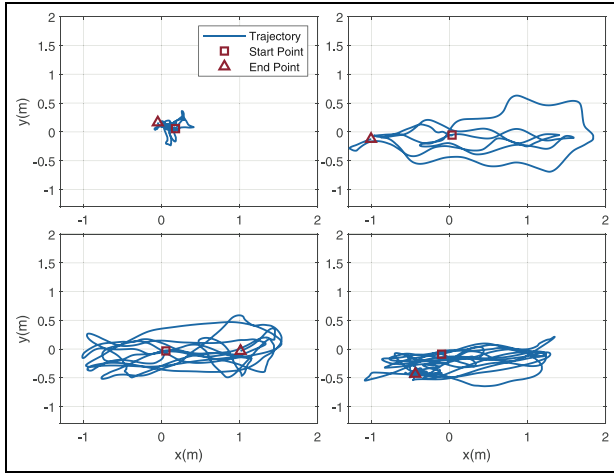


Figure 6. Examples of motion path capture by the OptiTrack system during the recording of the HELMET dataset. Figure shows the trajectories for the first 100 s of Sets 2 (top-left), 7 (top-right), 9 (bottom left) and 12 (bottom right).

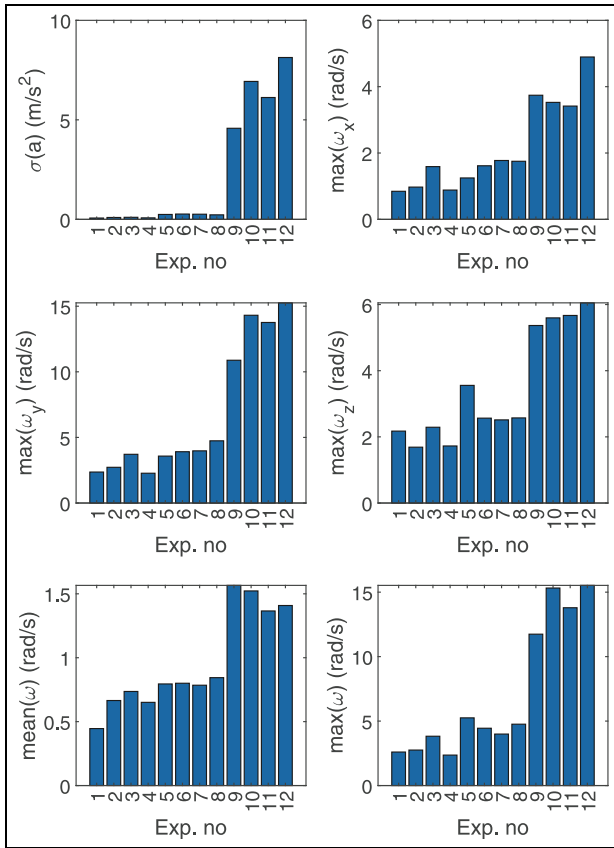


Figure 7. Inertial properties of the HELMET dataset. Top Left: Variance in acceleration measurements, Top Right: Maximum angular velocity along x-axis, Middle Left: Maximum angular velocity along y-axis, Middle Right: Maximum angular velocity along z-axis, Bottom Left: Mean angular velocity magnitude, Bottom Right: Maximum angular velocity magnitude.

in Figure 6. The variance of the accelerometer measurement, and maximum and mean values of the gyroscope measurements are shown in Figure 7. Variance in the

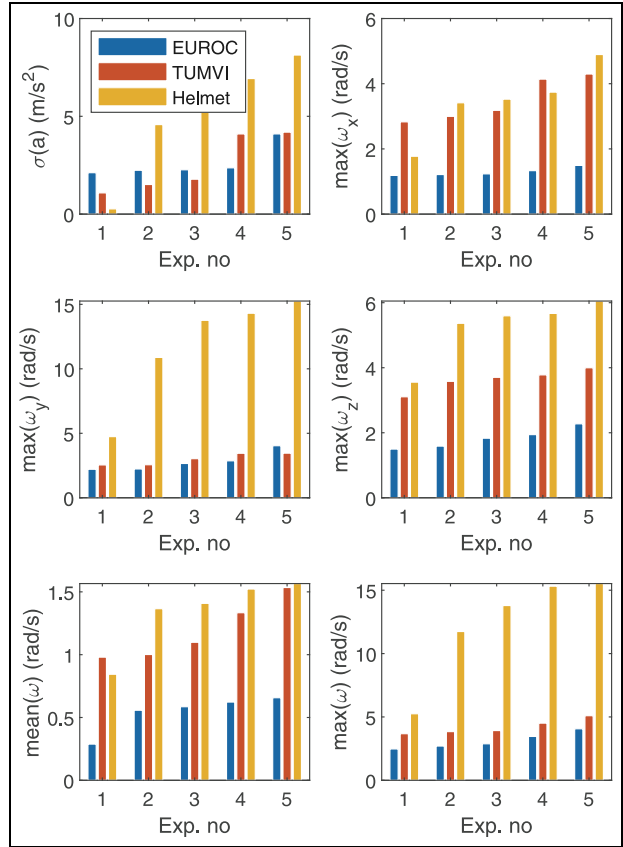


Figure 8. Comparison of EuRoC, TUM VI, and HELMET datasets. Top left: variance in acceleration; top right: maximum angular velocity along the x-axis, middle left: maximum angular velocity along the y-axis; middle right: maximum angular velocity along the z-axis; bottom left: mean angular velocity magnitude; bottom right: maximum angular velocity magnitude.

accelerometer data quantifies the acceleration due to non-gravitational forces on the helmet, which act as a disturbance during orientation estimation. Sets 1–4 have the lowest variance, while sets 9–12 have the highest variance caused by high acceleration forces during the sprints and the sudden stops. The sets also have different angular velocity profiles. A higher angular velocity results in a greater error during the integration step.

Compared to common IMU datasets considered in the literature that are based on quadcopter flight motion, such as EuRoC³² and TUM VI,³³ the HELMET dataset demonstrates that head motion is subject to higher angular velocities and accelerations. This is illustrated in Figure 8, which compares the angular velocity and acceleration data between EuRoC, TUM VI, and HELMET. Compared to the quadcopter-based datasets, the HELMET dataset has a higher yaw angular rate, which can result in high bias in the estimation and may require the use of magnetometer measurements for correcting the bias error. It is also noted that most neural network models in the literature developed using the EuRoC and TUM VI datasets do not use magnetometer measurements as input,

Table 2. Activation layer and loss.

No.	Activation			QAE
	First	Hidden	Last	
1	tanh	tanh	tanh	0.21678
2	PReLU	PReLU	PReLU	0.16618
3	PReLU	PReLU	tanh	0.19255
4	GELU	GELU	GELU	0.17178
5	GELU	GELU	tanh	0.16484

as magnetic field measurements are not reliable around the noise generated by the propeller motors. Therefore, models that are built on only the gyroscope and accelerometer signals experience a more pronounced drift in their estimate of yaw rotation due to the integration of the gyroscope noise. The HELMET dataset has been made available online at.⁴⁵

From the 12 experiments recorded in the HELMET dataset, 12 individual CNN model samples can be trained and evaluated using the leave-one-out cross-validation method. This method splits the set of recorded motion profile data into 11 training sets and one testing set. A CNN model is then trained on the training set data and tested on the one remaining set left out. This procedure is repeated until every set has been included in the testing set. The leave-one-out cross-validation method is used in the training of CNN models for the performance analysis to be discussed in Section 5.3. This procedure is time-consuming but provides better view of the generalizability of the network.

Activation functions and loss

The HELMET data is used to train and evaluate the proposed CNN-based head orientation estimation solution. Different combinations of the activation functions discussed in Section 4.2 were implemented for training the CNN model, and the average QAE (4) of the estimate from the testing data is presented in Table 2. As described in the table, all the hidden layers use the same activation function, while the activation function for the input and output layers were selected independently. A lower QAE value represents a more accurate estimation, and the best results were obtained with the GELU activation function in the first and the hidden layers and the tanh activation function in the output layer. The results in Table 2 are compiled using sets 2, 5, 9, and 12 from HELMET for testing and leaving the rest for training. This distribution of the dataset includes sets from different activities in the testing set and the resulting testing error is more representative of the expected error during deployment of the CNN model.

The loss function, as discussed in Section 4.3, is another hyper-parameter in training CNN models that influence the accuracy of the learned orientation estimation solution. A comparison of the average QAE for

Table 3. Loss function and QAE.

No.	Total Loss	Loss Function	Train QAE	Test QAE
1	L_1	$L^{(1)}, (6)$	0.3278	0.3542
2	L_2	$L^{(1)}, (6)$	0.4634	0.4703
3	L_2	$L^{(1)}, (7)$	0.4661	0.4844
5	L_1	$L^{(2)}$	0.3250	0.3489
6	L_2	$L^{(2)}$	0.6163	0.6325
7	L_1	$L^{(3)}$	0.2369	0.2545
8	L_2	$L^{(3)}$	0.2557	0.2669

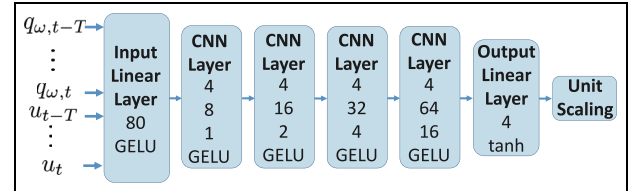


Figure 9. Convolutional Neural Network architecture used for orientation estimation. The parameters of the input and output layer from top to bottom are the output layer size and the activation function. The parameters of the CNN layer from top to bottom are the filter size, the number of filters, the dilation gap and the activation function.

models trained with different loss functions is presented in Table 3. The small differences between Train QAE and Test QAE for most of the tests in Table 3 show that the trained CNN model is not over-fitted to the training data and generalizes well for the testing data. For the total loss we used either L_1 or L_2 . In several applications, the loss L_1 is shown to be more robust to outliers in the data compared to L_2 .⁴⁶ This is also confirmed in our tests as L_1 produces less testing error compared to L_2 . The lowest average QAE for the testing data was obtained with L_1 as the total loss function and $L^{(3)}$ as the sample loss function.

The remaining hyper-parameters for the training of the CNN model, such as the dilation gap, filter size, and number of hidden layers, were chosen using a combination of Bayesian optimization³⁴ and trial and error. The final CNN model used for orientation estimation is shown in Figure 9. The inputs of the neural network are the accelerometer and magnetometer measurements and the quaternions obtained by integrating the gyroscope measurements. The number of past measurement samples used at the input were set to 32, which corresponds to 0.128 s of inertial data. Each inertial measurement is of size 10, which makes the total input features to equal 320.

Performance analysis

The proposed CNN-based method (3) is compared to ESKF and Mahony for estimating orientation of head motion. For the 12 trained model samples obtained from HELMET, Figure 10 presents the average QAE

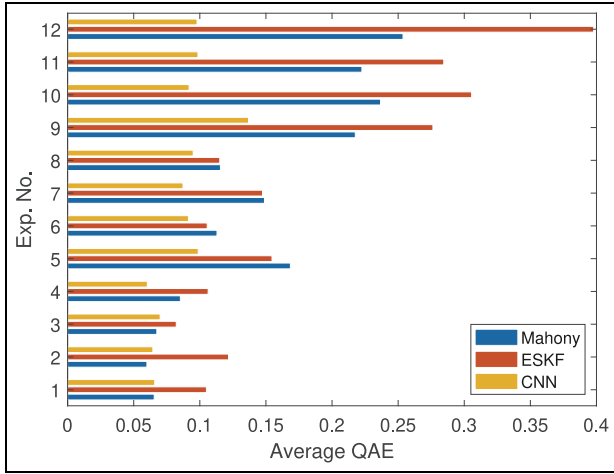


Figure 10. Error comparison of different estimation methods.

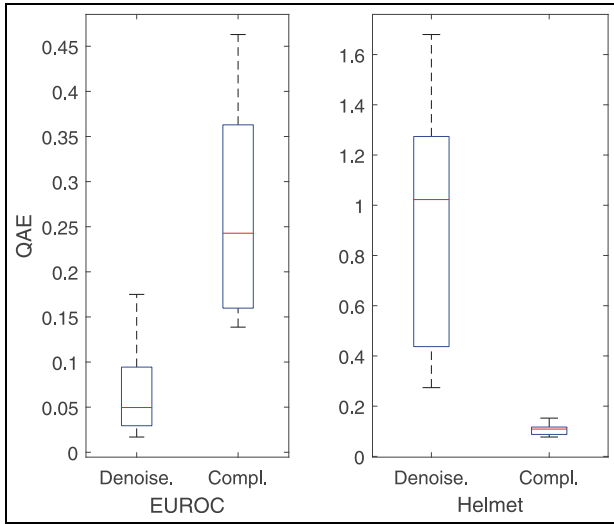


Figure 11. Comparison of QAE of denoising-based method and our complementary based method for EUROC and Helmet datasets.

quantifying the error between the orientation estimate and the true orientation. Overall, the CNN models show lower average estimation error compared to ESKF and Mahony. We can see that CNN models perform better for all motion profiles, except in Exp. 2 and 3, where the Mahony method slightly outperforms the proposed method. The proposed CNN-based solution significantly outperforms ESKF and Mahony when the magnitude of the acceleration disturbance is high (Exp. 9–12).

We also compared our method with the CNN-based method in Brossard et al.,²⁶ which removes noise from gyroscope data before integrating it to obtain orientation, and the error distributions are presented in Figure 11. For the comparison, models are trained and tested using both the EUROC dataset and the HELMET dataset. It is observed that the proposed method shows a lower average estimation error on the

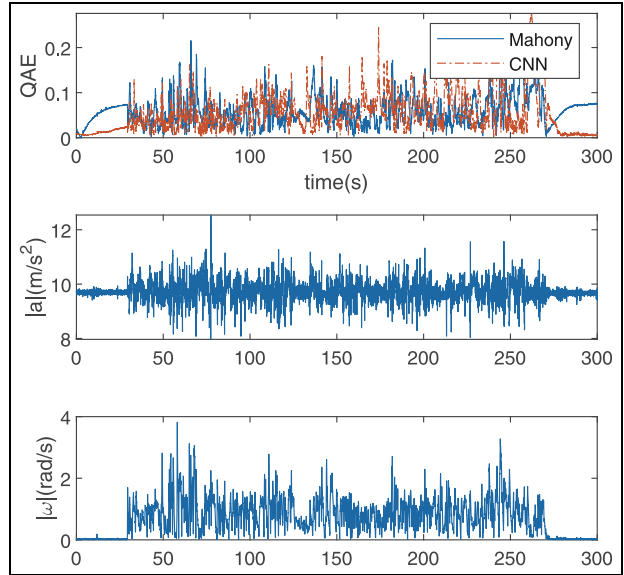


Figure 12. Comparison of QAE of Mahony and CNN for Exp. 3.

HELMET dataset while the denoising-based method performs better on the EUROC dataset. The contrast in performance is due to the difference in the motion profiles that the two datasets capture, and it illustrates the importance of application-specific orientation estimation solutions. The HELMET dataset captures faster yaw rotations and higher acceleration disturbance than the quadcopter-based EUROC database, and the estimation of the yaw angle in the proposed solution benefits from the corrections introduced by the accelerometer and magnetometer measurements.

Robustness

To improve the robustness of the CNN estimator we used sensor data that contained different levels of acceleration disturbances for training. As a result, CNN performs better than other techniques when disturbances are high at the cost of losing some accuracy under normal conditions. This can be seen in Figure 10 where the accuracy of CNN is greater for experiments with high acceleration disturbance (Exp. 9, 10, 11, and 12), while the average error from Mahony is slightly lower in Exp. 2 and 3. We can see in Figures 12 to 14 that, when the acceleration disturbance is low the QAE of our CNN-based solution is comparable to other methods, while under high disturbance the QAE of the proposed solution is significantly lower.

Conclusion

An integrated solution for head orientation estimation was presented in this work. The proposed solution used the inertial and magnetometer measurements from an instrumented helmet, and a CNN-based estimation algorithm was developed motivated by complementary filters. The CNN model was trained and evaluated on

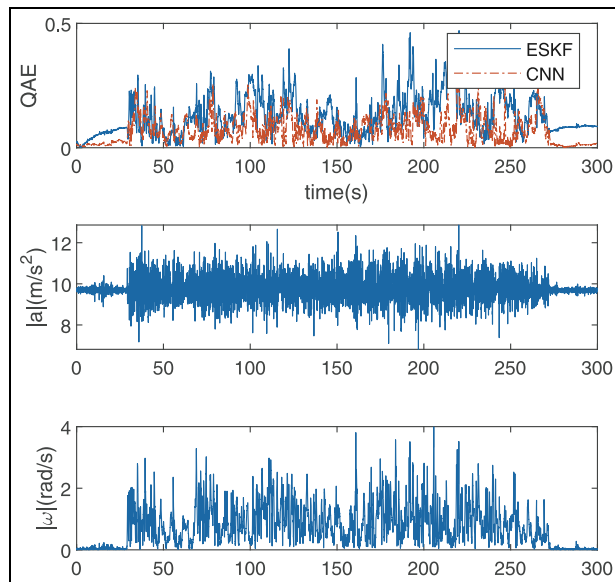


Figure 13. Comparison of QAE of Mahony and CNN for Exp. 7.

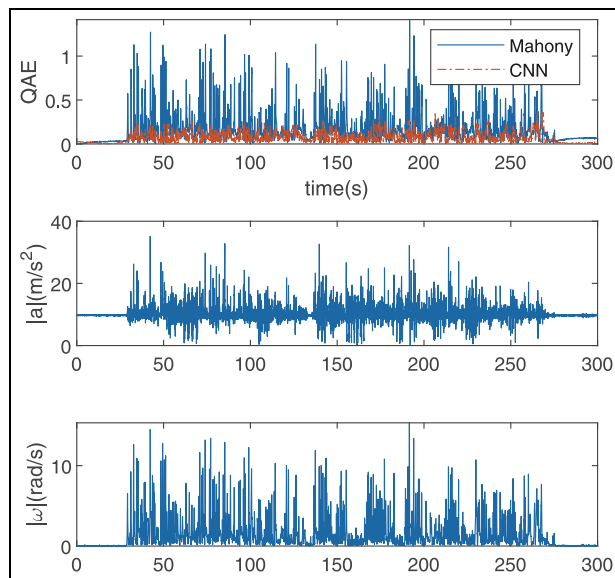


Figure 14. Comparison of QAE of Mahony and CNN for Exp. 10.

data collected to represent a wide range of head motion profiles. The selection of hyper-parameters for the CNN model was discussed in detail, and the final selection was optimized for the application considered in this work.

The proposed orientation estimation method was evaluated experimentally and compared to both learning and non-learning-based orientation estimation algorithms found in the literature for comparable applications. Test results prove the advantages of the proposed CNN-based solution, particularly for motion profiles with high acceleration disturbance that are characteristic of head motion.

It should be noted that the IMU measurements in the HELMET dataset capture a wide range of acceleration disturbances that are characteristic of head motion,

but the dataset does not consider variations in the magnetic disturbance level that may result from external sources of magnetic fields. Therefore, models trained from HELMET may display sensitivity to magnetic disturbances. Another possible limitation of HELMET is that the collected data is sampled at a constant sampling rate, and the effect of variable sampling rate is not considered in this study.

Future work is planned to evaluate the proposed solution in situations with intermittent periods of high magnetic disturbance, and investigate solutions to enhance the robustness of the estimation under these noisy magnetometer measurements. A possible solution may involve multiple CNN models that are trained to carry out the estimation of orientation at different levels of electromagnetic disturbance. New sensing systems and related algorithms could also be investigated to improve head orientation estimation and enable trajectory estimation. Finally, variable sampling frequency of the IMU measurements, and extensions of the proposed solution to accommodate for this change in the input data, are also to be included in future work.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the New Hampshire Innovation Research Center [grant number 13R307].

ORCID iD

Se Young Yoon  <https://orcid.org/0000-0002-3385-8447>

References

1. Seo YD and Hong KS. Calibration-free augmented reality in perspective. *IEEE Trans Vis Comput Graph* 2000; 6(4): 346–359.
2. Li P, Meziane R, Otis MJ, et al. A smart safety helmet using IMU and EEG sensors for worker fatigue detection. In: *Proceedings of 2014 IEEE international symposium on robotic and sensors environments (ROSE)*, pp.55–60. New York: IEEE.
3. Azuma R, Hoff B, Neely H, et al. A motion-stabilized outdoor augmented reality system. In: *Proceedings IEEE virtual reality (Cat. No. 99CB36316)*, pp.252–259. New York: IEEE.
4. Herath S, Yan H and Furukawa Y. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In: *2020 IEEE international conference on robotics and automation (ICRA)*, pp.3146–3152. New York: IEEE.
5. Shuster MD and Oh SD. Three-axis attitude determination from vector observations. *J Guid Control* 1981; 4(1): 70–77.

6. Wertz J. *Spacecraft attitude determination and control*. Dordrecht: Kluwer Academic Publishers, 1978.
7. Silveira G, Malis E and Rives P. An efficient direct approach to visual slam. *IEEE Trans Robot* 2008; 24: 969–979.
8. Michel T, Fourati H, Geneves P, et al. A comparative analysis of attitude estimation for pedestrian navigation with smartphones. In: *Proceedings of 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Banff, AB, Canada, 2015, pp.1–10.
9. Foxlin E. Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. In: *Proceedings of the IEEE 1996 virtual reality annual international symposium*, pp.185–194. New York: IEEE.
10. Göderer E. A quaternion-based unscented Kalman filter for orientation tracking. In: *Proceedings of the 6th international conference on information fusion*. New York: IEEE.
11. Madyastha V, Ravindra V, Mallikarjunan S, et al. Extended Kalman filter vs. error state kalman filter for aircraft attitude estimation. In: *AIAA guidance, navigation, and control conference*. Reston: AIAA.
12. Vitali RV, McGinnis RS and Perkins NC. Robust error-state Kalman filter for estimating IMU orientation. *IEEE Sens J* 2021; 21(3): 3561–3569.
13. Mahony R, Hamel T and Pfimlin JM. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans Automat Contr* 2008; 53: 1203–1218.
14. Markley FL. Attitude determination using vector observations and the singular value decomposition. *J Astron Sci* 1988; 36(3): 245–258.
15. Mortari D, Markley FL and Singla P. Optimal linear attitude estimator. *J Guid Control Dyn* 2007; 30(6): 1619–1627.
16. Wu J, Zhou Z, Gao B, et al. Fast linear quaternion attitude estimator using vector observations. *IEEE Trans Autom Sci Eng* 2018; 15(1): 307–319.
17. Madgwick SOH, Harrison AJL and Vaidyanathan R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In: *2011 IEEE international conference on rehabilitation robotics*, pp.1–7. New York: IEEE.
18. Smola AJ and Schölkopf B. A tutorial on support vector regression. *Stat Comput* 2004; 14(3): 199–222.
19. Yan H, Shan Q and Furukawa Y. *RIDI: Robust IMU double integration*. Technical report, ArXiv, 2018, <https://arxiv.org/abs/1712.09004> (accessed October 13, 2022).
20. Chen C, Lu X, Markham A, et al. *Ionet: Learning to cure the curse of drift in inertial odometry*. Technical report, ArXiv, 2018. <https://arxiv.org/abs/1802.02209> (accessed October 13, 2022).
21. Wagstaff B and Kelly J. Lstm-based zero-velocity detection for robust inertial navigation. In: *Proceedings of 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nantes, France, 2018, pp.1–8.
22. Brossard M, Barrau A and Bonnabel S. *Rins-w: Robust inertial navigation system on wheels*. Technical Report arXiv:1903.02210, ArXiv, 2020.
23. Chiang KW, Chang HW, Li CY, et al. An artificial neural network embedded position and orientation determination algorithm for low cost mems INS/GPS integrated sensors. *Sensors* 2009; 9(4): 2586–2610.
24. Weber D, Gühmann C and Seel T. *Neural networks versus conventional filters for inertial-sensor-based attitude estimation*. Technical report, ArXiv, 2020, <http://arxiv.org/abs/2005.06897> (accessed October 13, 2022).
25. Krizhevsky A, Sutskever I and Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM* 2017; 60(6): 84–90.
26. Brossard M, Bonnabel S and Barrau A. Denoising IMU gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robot Autom Lett* 2020; 5(3): 4796–4803.
27. Abolfazli Esfahani M, Wang H, Wu K, et al. Aboldepio: A novel deep inertial odometry network for autonomous vehicles. *IEEE Trans Intell Transp Syst* 2020; 21(5): 1941–1950.
28. Esfahani MA, Wang H, Wu K, et al. Orinet: Robust 3-d orientation estimation with a single particular IMU. *IEEE Robot Autom Lett* 2020; 5(2): 399–406.
29. Zhao X, Deng C, Kong X, et al. Learning to compensate for the drift and error of gyroscope in vehicle localization. In: *2020 IEEE intelligent vehicles symposium (IV)*, pp.852–857. New York: IEEE.
30. Zhao L, Pingali G and Carlbom I. Real-time head orientation estimation using neural networks. In: *Proceedings of the International Conference on Image Processing*, Rochester, NY, USA, 2002, volume 1. pp.I–I.
31. Liu J, Gu Y and Kamijo S. Joint customer pose and orientation estimation using deep neural network from surveillance camera. In: *2016 IEEE international symposium on multimedia (ISM)*, pp.216–221. New York: IEEE.
32. Burri M, Nikolic J, Gohl P, et al. The Euroc micro aerial vehicle datasets. *Int J Rob Res* 2016; 35(10): 1157–1163.
33. Schubert D, Goll T, Demmel N, et al. The tum vi benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp.1680–1687. New York: IEEE.
34. Shahriari B, Swersky K, Wang Z, et al. Taking the human out of the loop: A review of Bayesian optimization. *Proc IEEE* 2016; 104: 148–175.
35. Ioffe S and Szegedy C. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. Technical report, ArXiv, 2015, <http://arxiv.org/abs/1502.03167> (accessed October 13, 2022).
36. Ramachandran P, Zoph B and Le QV. *Searching for activation functions*. Technical report, ArXiv, 2017, <http://arxiv.org/abs/1710.05941> (accessed October 13, 2022).
37. Kaiming H, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*, pp.1026–1034. New York: IEEE.
38. Huynh DQ. Metrics for 3d rotations: comparison and analysis. *J Math Imaging Vis* 2009; 35(2): 155–164.
39. Liu L, Jiang H, He P, et al. *On the variance of the adaptive learning rate and beyond*. Technical report arXiv:1908.03265 [cs, stat], ArXiv, 2019, <http://arxiv.org/abs/1908.03265> (accessed October 13, 2022).
40. Zhang MR, Lucas J, Hinton G, et al. *Lookahead optimizer: k steps forward, 1 step back*. Technical report arXiv:1907.08610, ArXiv, 2019, <http://arxiv.org/abs/1907.08610> (accessed October 13, 2022).
41. Shorten C and Khoshgoftaar TM. A survey on image data augmentation for deep learning. *J Big Data* 2019; 6: 60. <https://doi.org/10.1186/s40537-019-0197-0>

42. Zampella F, Khider M, Robertson P, et al. Unscented Kalman filter and magnetic angular rate update (maru) for an improved pedestrian dead-reckoning. In: *Proceedings of the 2012 IEEE/ION position, location and navigation symposium*, pp.129–139. New York: IEEE.
43. VectorNav. Vectornav's vn-100 imu/ahrs, the world's most trusted surface mount solution, <https://www.vectornav.com/products/detail/vn-100> (accessed 9 May 2022).
44. Optitrack. Flex 13 - an affordable motion capture camera, <https://www.optitrack.com/cameras/flex-13/indepth.html> (accessed 9 May 2022).
45. Zaheer H. Helmet mounted inertial measurement unit dataset, <https://kaggle.com/datasets/muhammadhamadzaheer/helmet-mounted-inertial-measurement-unit-dataset> (accessed 9 May 2022).
46. Zhang Z. Parameter estimation techniques: a tutorial with application to conic fitting. *Image Vis Comput* 1997; 15(1): 59–76.